ML Workshops
A Subsidiary of Solivar Labs

Solivar Labs

K-MEANS

UNSUPERVISED LEARNING

Overview

K-means (MacQueen, 1967) is one of the simplest unsupervised learning algorithms that solve the well known clustering problem.

The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result.

The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

Lloyd's Algorithm:

Lloyd algorithm is the most popular way of implementing k-means.

- *First we choose **k** - the number of clusters we want*
- *Then we randomly initialize **k** cluster centers (cluster centroids)*

This is an iterative algorithm, and on each iteration it does 2 things

- *Cluster assignment step*
- *Move centroids step*

Cluster Assignment Step:

- *Go through each example and choose the closest centroids*
- *and assign the example to it*

Move Centroids Step:

- *Calculate the average for each group*
- *and move the centroids there*

ML Workshops
A Subsidiary of Solivar Labs

Pseudo Code:

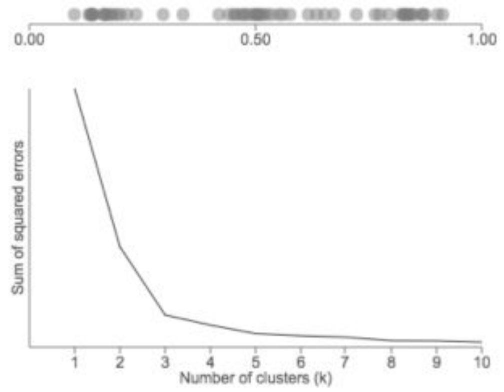
```
k-means(k, $\{x_i\}$):

● Randomly initialize k cluster centroids $\mu=(\mu_1,\mu_2,\ldots,\mu_k)\in\mathbb{R}k+1$
● Repeat:
    ○ Cluster assignment step:

        i.  for i=1 to m:
        ii. $c_i \leftarrow$ closest to $x_i$ centroid using **Euclidean Distance**

                    $Dist = \|x_i - \mu_i\|^2$
    ○ Move centroids step:
        i.  for i=1 to k:
        ii. $\mu_k \leftarrow$ average of all points assigned to $c_k$
```

SOLIVAR LABS

Choosing the number of clusters (value of k):

There are two ways to choose the value of k

- *Manually - by looking at the dataset*
- *Using Elbow Method*



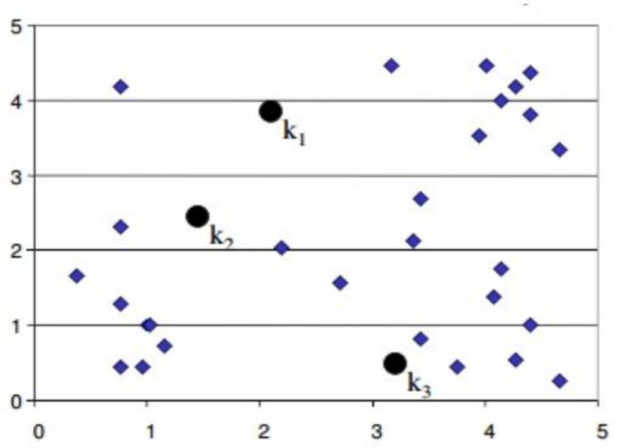
Elbow Method:

*The **Elbow method** looks at the percentage of variance explained as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't give much better modeling of the data. More precisely, if one plots the percentage of variance explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the "elbow criterion".*

Example:

Step-1:

Start with number of clusters we want e.g., 3 in this case. K-Means algorithm start the process with random centers in data, and then tries to attach the nearest points to these centers.

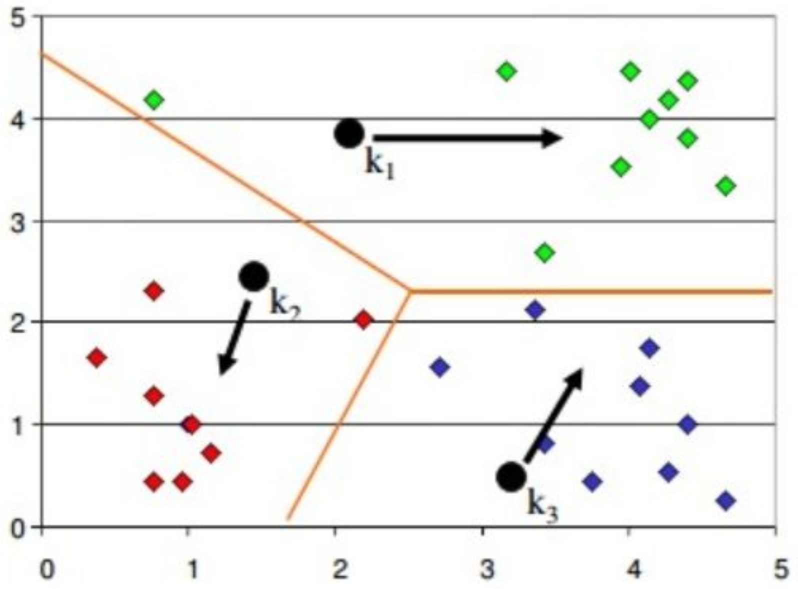


ML Workshops
A Subsidiary of Solivar Labs

SOLIVAR LABS

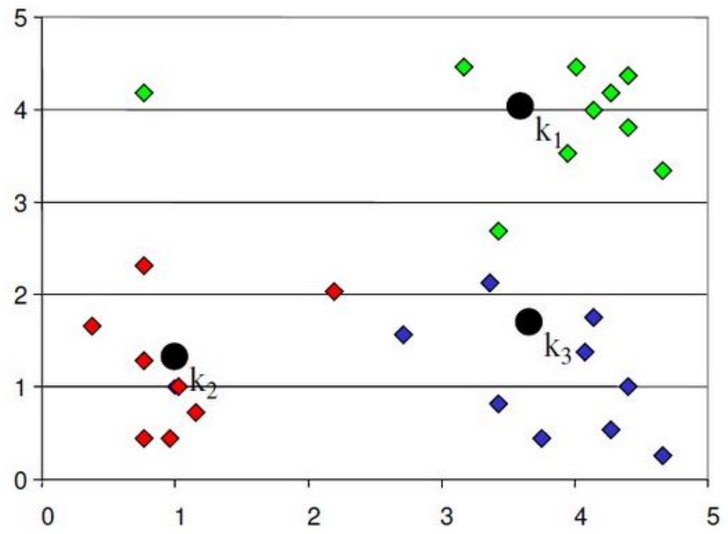
Step-2:

Algorithm then moves the randomly allocated centers to the means of created groups.



Step-3:

In the next step, data points are again assigned to these newly created centers.



Steps 2 & 3 are repeated until no member changes their association/ groups

Pros & Cons

Pros

- *Practically work well even some assumptions are broken;*
- *Simple, easy to implement;*
- *Easy to interpret the clustering results;*
- *Fast and efficient in terms of computational cost, typically O(K*n*d)*

Cons

- ***Uniform effect:** Often produce clusters with relatively uniform size even if the input data have different cluster size;*
- ***Spherical assumption hard to satisfy:** Correlation between features break it, would put extra weights on correlated features(should take actions depending on the problems); cannot find non-convex clusters or clusters with unusual shapes;*
- ***Different densities:** May work poorly with clusters with different densities but spherical shape;*